ESANN 2013 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2013, i6doc.com publ., ISBN 978-2-87419-081-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100131010.

Learning Regression Models with Guaranteed Error Bounds

Clemens Otte *

Siemens AG, Corporate Technology
Otto-Hahn-Ring 6, 81739 Munich, Germany
clemens.otte@siemens.com

Abstract. The combination of a symbolic regression model with a residual Gaussian Process is proposed for providing an interpretable model with improved accuracy. While the learned symbolic model is highly interpretable the residual model usually is not. However, by limiting the output of the residual model to a defined range a worst-case guarantee can be given in the sense that the maximal deviation from the symbolic model is always below a defined limit. When ranking the accuracy and interpretability of several different approaches on the SARCOS data benchmark the proposed combination yields the best result.

1 Introduction

Consider the problem of learning a regression function $f : \mathbb{R}^p \rightarrow \mathbb{R}$ from n training examples $(\mathbf{x}_i, y_i)$, $i = 1, \ldots, n$. Deploying such a data-driven model in applications where incorrect model outputs may have fatal consequences requires ensuring that the model is correct for all possible inputs.

In practice, training data are almost always limited and may not represent all relevant operating conditions. Thus, it is crucial to understand what the model has learned and in particular how it will extrapolate to unseen data.

Traditionally, approaches like CART [1] or rule-based methods are considered as being interpretable. Yet this is only true as long as the number of nodes or rules is rather small, which means that the model is quite coarse. A possible compromise is to partition the input space similar to CART or rule learners but to use more complex submodels in the different partitions of the input space. MARS (multivariate adaptive regression splines) [2] and an approach called GUIDE (generalized unbiased interaction detection and estimation) [3] follow this strategy; both are briefly compared in an experiment in this paper.

In statistics, additive models [4] are often applied when the model shall be interpretable. The idea is to learn a multivariate function having p inputs as a sum of p univariate functions, that is, $y = \alpha + \sum_{i=1}^{p} f_i(x_i) + \epsilon$. The advantage is that the univariate functions may be easier to interpret. However, interactions between variables cannot be modeled.

In this paper a different strategy is proposed similar to an additive model in the sense that two functions are added in the final model. The first function is

*Work partially funded by German Federal Research Ministry, BMBF grant ALICE 01, IB10003 A-C.

ESANN 2013 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2013, i6doc.com publ., ISBN 978-2-87419-081-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100131010.

used as an analytical model learned by symbolic regression. Symbolic regression [5] seeks for a symbolic representation (i.e. an equation) best matching the given data. The learned analytical model is easily interpretable but has only moderate accuracy. Thus, a second function is learned on the residuals of the first function in order to improve the accuracy. The model of the second function is not interpretable. But by limiting its output to a defined range a worst-case guarantee can be given in the sense that the maximal deviation from the analytical model is always below a defined limit.

Related work: The idea of combining analytical models with data-driven models can be traced back to *hybrid neural networks* considered mainly in the 1990-ties, e.g. [6]. However, there are two key differences. First, in this paper the analytical model is not considered as being given but is learned from data by symbolic regression. Second, Gaussian Processes are used as residual models, facilitating the model selection as there is no need to empirically decide on appropriate neural network architectures as the model is mainly given by the data itself.

2 Safe and Interpretable Regression

Consider the combined model

$$f(\mathbf{x}) = a(\mathbf{x}) + g_\gamma(r(\mathbf{x})) \tag{1}$$

where a is an analytical (that is, fully interpretable and verified) model, r is a residual model and g_γ provides a limitation. We use $g_\gamma(x) = \max(\min(x, \gamma), -\gamma)$, $\gamma \in \mathbb{R}^+$, allowing changes of $\pm\gamma$ to the analytical model. The additive combination is chosen because it facilitates the training of the submodels and the interpretation of the overall model. If safety requirements allow trading off interpretability against accuracy, more complex combination schemes may be devised. For instance, uncertainty estimates of r could be used to influence γ, giving the residual model more impact in high-confidence regions.

In this paper a is learned using the Eureqa tool for symbolic regression[1], where the training data are used to search for an explicit symbolic representation based on genetic programming [5]. Alternatively, any method providing a fully interpretable model may be used.

The residual model r is learned as a *Gaussian Process* (GP), which is a linear smoother using a weighted average of the stored training outputs $\mathbf{y}$ to predict the output for a test input. It can be seen as a linear combination of n kernel functions, each one centered on one of n training points [7],

$$r(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i k(\mathbf{x}_i, \mathbf{x}), \quad \alpha = (\mathbf{K} + \sigma_n^2 \mathbf{I})^{-1} \mathbf{y} \tag{2}$$

where $\mathbf{x}$ is a test input, $\mathbf{x}_i$ are the training inputs and α is a weight vector with the $(n \times n)$ kernel matrix $\mathbf{K}$ and a hyperparameter σ_n.

[1] http://creativemachines.cornell.edu/eureqa

ESANN 2013 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2013, i6doc.com publ., ISBN 978-2-87419-081-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100131010.

On larger datasets (e.g. $n \gg 5000$ samples) $\mathbf{K}$ is not invertible due to time and memory limitations and approximations to (2) must be used. A good approximation is known as the *Subset-of-Regressors* (SR) method [7, p. 176]. Instead of summing over all n samples as in (2), only a subset of $m < n$ samples is used. Information provided by the full set of the n samples is exploited by considering the similarities between the selected m samples and the n samples in the form of an $(m \times n)$ matrix $\mathbf{K}_{mn}$.

Using the SR method the residual model is given as

$$r(\mathbf{x}) = \mathbf{k}_m(\mathbf{x})^T(\mathbf{K}_{mn}\mathbf{K}_{nm} + \sigma_n^2\mathbf{K}_{mm})^{-1}\mathbf{K}_{mn}\mathbf{y} \tag{3}$$

where $\mathbf{k}_m(\mathbf{x})$ is a vector describing the similarities between the test input $\mathbf{x}$ and the m training samples; the matrices $\mathbf{K}_{mn}$ and $\mathbf{K}_{nm} = (\mathbf{K}_{mn})^T$ describe the similarities between the m and n samples; and the $(m \times m)$ matrix $\mathbf{K}_{mm}$ describes the similarities among the m samples.

The approach is evaluated in the following experiment.

3 Experiment: SARCOS Benchmark

The objective of the experiment is to compare different regression approaches in terms of their accuracy and interpretability. We consider learning the inverse dynamics of a seven degrees-of-freedom SARCOS robot arm. The task is to map from a 21-dimensional input space to a joint torque, that is, we learn a function $f : \mathbb{R}^{21} \to \mathbb{R}$. There are 44484 training examples and 4449 test examples[2]. All 21 inputs x_i have been standardized to zero mean and standard deviation 1. The output y has zero mean. Results are given as *standardized mean squared error* (SMSE), which is the mean squared error on the test set divided by the variance of the target values in the test set.

We use the squared exponential kernel where each input dimension is scaled by an individual factor, allowing the down-weighting of irrelevant inputs. The kernel function is given as

$$k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{x}')^T D^{-2}(\mathbf{x} - \mathbf{x}')\right)$$

with diagonal matrix $D = \mathrm{diag}(\ell_1, \ldots, \ell_{21})$ containing the scaling factors of the input dimensions. In total 23 hyperparameters $(\ell_1, \ldots, \ell_{21}, \sigma_f, \sigma_n)$ are optimized during training and the same values are taken in both methods, GP and SR.

As the training set to too large to invert the kernel matrix in (2) it is clustered into 4000 clusters by *kmeans*. For each cluster center the nearest training sample is taken together with its output value, yielding 4000 training samples. For the GP method these 4000 samples are used in (2), where $n = 4000$. For the SR method in (3) the 4000 samples provide the subset, that is, $m = 4000$ and $n = 44484$. The experiments are repeated 10 times with different *kmeans* clusters and the mean and standard deviation of the test error over the 10 runs are reported.

[2]Data are publicly available from http://www.gaussianprocess.org/gpml/data/

ESANN 2013 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2013, i6doc.com publ., ISBN 978-2-87419-081-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100131010.

3.1 Results

Test set errors of different methods are reported in Table 1. The selection of the methods is based on their claim of providing interpretable models – except for GP and SR, which were chosen as representatives of nonparametric black-box approaches allowing high accuracy but no interpretability.

RBD refers to a physics-based model derived from rigid-body dynamics. The RBD result is taken from [7], p. 24. The RBD accuracy is rather poor, which may be explained by the fact that the robot is actuated hydraulically and is rather lightweight, so some rigid-body assumptions seem to be violated.

LR refers to a linear regression model with all 21 inputs. Alternatively, LASSO [4] or elastic net [4] could have been used in order to reduce the number of inputs. However, their modeling capabilities are the same as LR.

GUIDE [3] is a recursive partitioning algorithm creating a regression tree. Unlike CART, in GUIDE the leaves may contain linear models either with all inputs or with a subset of inputs. With the complexity of the submodels in the leaves being increased, the overall number of nodes can often be kept smaller, improving the interpretability of the model. The result in Table 1 is achieved with a model consisting of 35 linear models each having 21 variables.

MARS [2] builds the model as a linear combination of basis functions in the form of univariate splines and products of univariate splines. A key property of the basis functions is that they are zero over some part of their range, making it possible for them to operate locally. The result in Table 1 refers to a model with 21 basis functions where the maximum interaction level has been limited to 2, thus, only allowing pairwise products. The model can be found in [8]; it is not easy to interpret.

Eureqa [5] uses genetic programming for exploring a search space of equations assembled from pre-specified "building blocks" (e.g. arithmetic operators, sin, exp) to find an equation best matching the data. Using the mean-squared error to guide the search on the complete training set and taking the model having the smallest training set error yields:

$$\begin{aligned} y = {} & x_2 + 25.31\,x_{15} + 11.08\,x_1 + 11.08\,x_{18} + 1.73\,x_{21} + x_1x_{15} + \\ & x_4x_{21} - x_{11} - x_2x_{15} - x_4x_{18} - 1.73\,x_8x_9 \end{aligned} \tag{4}$$

This representation is sparser and more interpretable than the MARS model. Unlike the LR model, the Eureqa model includes several terms with two interacting variables, which explains its lower test set error in comparison to the LR model as shown in Table 1. While in this specific application (4) could in principle also be found by linear regression –assuming product terms are provided–, it should be stressed that genetic programming offers much more flexibility, going far beyond linear regression.

GP and **SR** refer to the methods applied on 4000 training samples selected with *kmeans* as described above. The corresponding entries in Table 1 show the results for learning the original problem and not the residuals.

ESANN 2013 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2013, i6doc.com publ., ISBN 978-2-87419-081-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100131010.

Method	SMSE	Interpretability
RBD	0.104	++
LR	0.075	+
GUIDE	0.033	o
MARS	0.059	o
Eureqa	0.062	++
GP	0.0183 $\pm$0.0006	–
SR	0.0111 $\pm$0.0003	–
Eureqa + SR ($\gamma = 5$)	0.0222 $\pm$0.0001	+
Eureqa + SR ($\gamma = 10$)	0.0132 $\pm$0.0002	+
Eureqa + SR ($\gamma = 15$)	0.0109 $\pm$0.0002	o
Eureqa + SR ($\gamma = \infty$)	0.0099 $\pm$0.0002	–

Table 1: Results on the inverse dynamics problem. The error is given as standardized-mean-squared error (SMSE) on the test set. γ refers to the limitation used in Eq. (1). GP and SR results show mean and standard deviation over 10 runs. The third column refers to the interpretability of the model: ++: very high, +: high, o: moderate, –: not interpretable.

Eureqa + SR refers to the combined model in (1), where SR learns the residuals. The limit $\gamma \in \mathbb{R}$ provides a worst-case guarantee on the error of the overall model. When choosing the limit, there is a trade-off between possible accuracy gains and the worst-case guarantee. In the SARCOS training set the output is in the range $[-108.5, 107.7]$, so the limits were chosen as $\gamma \in \{5, 10, 15, \infty\}$. For example, $\gamma = 10$ means that the maximal possible deviation from the analytical model is ± 10, which is less than ten percent of the maximal output values. The unlimited case is denoted as $\pm\infty$. Note that the tightest limit ± 5 already improves the accuracy considerably in comparison to the pure Eureqa model.

3.2 Discussion

The combination of the symbolic Eureqa model with an unlimited residual SR model yields the best accuracy, which is even slightly higher than that of the pure GP and SR models. However, in terms of safety nothing would be gained as it is not possible to fully verify the residual SR model – so the overall model output may be arbitrarily wrong in the worst case. Thus, the influence of the residual model has to be limited for providing a worst-case guarantee in the sense that the maximal deviation from the verifiable analytical model is always below that limit.

An important observation is that even a strongly limited SR model (e.g. ± 5) greatly improves the accuracy in comparison to the pure Eureqa model, making it possible to boost the accuracy while still having rather tight limits around the verifiable model.

ESANN 2013 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2013, i6doc.com publ., ISBN 978-2-87419-081-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100131010.

4 Conclusions

Understanding what a model has learned and in particular how it will extrapolate to unseen data becomes a crucial concern if the model correctness must be ensured not only for the available data but for all possible input combinations.

Approaches traditionally considered as interpretable, like MARS, CART or GUIDE, particularly suffer from the trade-off between sparsity (i.e. interpretability) and accuracy: for example, a GUIDE model trained on the SARCOS benchmark with 35 linear models is not really interpretable but sparser GUIDE models have a worse accuracy.

An appealing idea is thus to split the problem into learning a sparse, fully verifiable model and then learning a residual model to enhance the accuracy of the first. The residual model is not required to be fully interpretable but its influence is limited to a predefined range. Thus, the combined model has an improved accuracy and provides error bounds in the sense that the deviation from the verifiable model is always below a defined limit.

Using symbolic regression for learning the sparse and fully verifiable model has proven beneficial in the SARCOS example considered here. The model is sparser and easier to interpret than models provided by MARS, CART or GUIDE.

Learning the residuals by Gaussian Processes is appealing as it reduces the training effort to choosing the kernel function and optimizing the hyperparameters by gradient descent. This can be easier than learning the residuals by a multi-layer perceptron whose architecture is usually determined by trial-and-error.

References

[1] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Statistics/Probability Series. Wadsworth Publishing Company, Belmont, California, U.S.A., 1984.

[2] Jerome H. Friedman. Multivariate Adaptive Regression Splines. *The Annals of Statistics*, 19(1):1–67, 1991.

[3] Wei-Yin Loh. Regression by parts: Fitting visually interpretable models with guide. In *Handbook of Data Visualization*, Springer Handbooks Comp.Statistics, pages 447–469. 2008.

[4] T. Hastie, R. Tibshirani, J. Friedman, and J. Franklin. *The elements of statistical learning: data mining, inference and prediction*. Springer, 2009.

[5] Michael Schmidt and Hod Lipson. Distilling Free-Form Natural Laws from Experimental Data. *Science*, 324(5923):81–85, April 2009.

[6] M. Schlang, B. Feldkeller, B. Lang, T. Poppe, and T. Runkler. Neural computation in steel industry. In *Proceedings of European Control Conference '99*, pages 1–6, Karlsruhe, Germany, 1999. Verlag rubicon.

[7] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

[8] Clemens Otte. Safe and interpretable machine learning: A methodological review. In *Computational Intelligence in Intelligent Data Analysis*, volume 445, pages 111–122. Springer, 2013.